

**PROPUESTA DE UN CLASIFICADOR DEL GRADO DE MADUREZ DEL
AGUACATE HASS, MEDIANTE UNA ARQUITECTURA DE REDES NEURONALES
CONVOLUCIONALES**

JOSÉ ALEJANDRO MORENO AGUDELO

UNIVERSIDAD TECNOLÓGICA DE PEREIRA

FACULTAD DE INGENIERÍAS

PROGRAMA DE MAESTRÍA EN INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

PEREIRA-2020

**PROPUESTA DE UN CLASIFICADOR DEL GRADO DE MADUREZ DEL AGUACATE
HASS, MEDIANTE UNA ARQUITECTURA DE REDES NEURONALES
CONVOLUCIONALES**

JOSÉ ALEJANDRO MORENO AGUDELO

**TRABAJO DE GRADO PARA OPTAR POR EL TÍTULO DE MAGISTER EN
SISTEMAS Y COMPUTACIÓN**

DIRECTOR

Ph. D GUILLERMO ROBERTO SOLARTE MARTINEZ

UNIVERSIDAD TECNOLÓGICA DE PEREIRA

FACULTAD DE INGENIERÍAS

PROGRAMA DE MAESTRÍA EN INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

PEREIRA-2020

Nota de Aceptación

Presidente del Jurado

Jurado

Jurado

AGRADECIMIENTOS

Agradezco inmensamente a las dos personas a quienes debo este logro, al Doctor Jaime Ángel y a mi madre, quienes fueron siempre un apoyo, y me incentivaron para emprender el viaje en Italia, siempre siendo un motivo para esforzarme cada vez más y sobrepasar los momentos difíciles.

De igual manera, agradezco a mis amigos italianos, quienes me brindaron una amistad incondicional, y me permitieron conocer su hermosa y rica cultura, compartiendo mis mejores años recientes.

Agradezco igualmente al Doctor Guillermo Solarte, quien aceptó desde un primer momento dirigir este proyecto.

Por último, agradezco a Dios y a la virgen por haberme permitido vivir esta maravillosa experiencia y por siempre guiarme en los mejores caminos.

TABLA DE CONTENIDO

INTRODUCCIÓN	8
DEFINICIÓN DEL PROBLEMA.....	11
JUSTIFICACIÓN	12
ALCANCE	13
OBJETIVOS	14
OBJETIVO GENENERAL.....	14
OBJETIVOS ESPECIFICOS	14
HIPÓTESIS.....	14
METODOLOGÍA.....	15
TIPO DE INVESTIGACIÓN	15
MÉTODO DE INVESTIGACIÓN.....	15
TÉCNICAS PARA LA RECOLECCIÓN DE INFORMACIÓN	16
ESTUDIO DE CAMPO.....	17
MARCO DE REFERENCIA	21
APRENDIZAJE DE MÁQUINAS.....	22
RED NEURONAL	23
RED NEURONAL CONVOLUCIONAL.....	24
DETECCIÓN DE OBJETOS.....	25
MÁQUINAS DE SOPORTE VECTORIAL.....	26
ESTADO DEL ARTE	28
DESARROLLO DE ALGORITMO Y PROTOTIPO MÓVIL PARA MEDIR EL GRADO DE MADUREZ DEL AGUACATE HASS MEDIANTE PROCESAMIENTO DIGITAL DE IMÁGENES.	28
RIPENESS CLASSIFICATION OF BANANAS USING AN ARTIFICIAL NEURAL NETWORK.....	29
IMAGE PROCESSING APPLIED TO CLASSIFICATION OF AVOCADO VARIETY HASS (PERSEA AMERICANA MILL.) DURING THE RIPENING PROCESS.....	31
VEHICLE COLOR CLASSIFICATION UNDER DIFFERENT LIGHTING CONDITIONS THROUGH COLOR CORRECTION	32
VEHICLE COLOR RECOGNITION USING CONVOLUTIONAL NEURONAL NETWORK ...	33
COLOUR RECOGNITION IN IMAGES USING NEURAL NETWORKS.....	34

SKIN COLOR DETECTION MODEL USING NEURAL NETWORKS AND ITS PERFORMANCE EVALUATION	36
AUTOMATIC RECOGNITION OF RIPENING TOMATOES BY COMBINING MULTI-FEATURE FUSION WITH A BI-LAYER CLASSIFICATION STRATEGY FOR HARVESTING ROBOTS	40
PREDICTION OF BANANA QUALITY INDICES FROM COLOR FEATURES USING SUPPORT VECTOR REGRESSION	41
FRUIT RIPENESS ESTIMATION FOR AVOCADO USING THERMAL IMAGING.....	43
APPLE RIPENESS ESTIMATION USING ARTIFICIAL NEURAL NETWORK	44
TOMATO CLASSIFICATION ACCORDING TO ORGANOLEPTIC MATURITY (COLORATION) USING MACHINE LEARNING ALGORITHMS K-NN, MPL, AND K-MEANS CLUSTERING	45
DESCRIPCIÓN DE LOS RESULTADOS Y LA ARQUITECTURA	47
DESCRIPCIÓN DE LA ARQUITECTURA PROPUESTA	47
RESULTADOS	65
CONCLUSIONES	70
TRABAJOS FUTUROS.....	84
BIBLIOGRAFÍA	85

INDICE DE ILUSTRACIONES

Ilustración 1.....	18
Ilustración 2.....	19
Ilustración 3.....	19
Ilustración 4.....	20
Ilustración 5.....	48
Ilustración 6.....	49
Ilustración 7.....	65
Ilustración 8.....	66
Ilustración 9.....	66
Ilustración 10	67
Ilustración 11	67

INDICE DE TABLAS

Tabla 1.....	68
Tabla 2.....	69

INTRODUCCIÓN

Según el ministerio de agricultura de Colombia¹, se estima que en este país existe una cantidad aproximada de 13.000 productores en cerca de 18.000 unidades productivas, que a su vez tienen como actividad económica principal el cultivo de aguacate.

Entre todas sus variedades, el aguacate Hass representa además un aproximado del 25% sobre la totalidad del área sembrada con aguacate en el país.

De esta actividad, se estima, además, que cerca de 54.000 personas en el país se ven involucradas de forma directa o indirecta en las distintas etapas de la cadena productiva del aguacate.

En este sentido, el ministerio presenta cifras en donde se muestra que, en los últimos 5 años, la producción del aguacate ha crecido en un 89%, y su rendimiento en un 17%.

A nivel mundial, Colombia logró una posición envidiable, alcanzando el tercer puesto en el cultivo del aguacate en el año 2018, en relación con el área cosechada, con un 6% que representan un aproximado de más de 54.000 hectáreas, y cuarto lugar con respecto al total de la producción, con un 11% del total, que representa una cantidad aproximada de más de 540.000 toneladas.

¹ Ministerio colombiano de agricultura, 2019, "Cadena de aguacate, indicadores e instrumentos".

Sin embargo, en promedio, los últimos 6 años se ha caracterizado con una tendencia de crecimiento en los costos de producción, de alrededor de un 5%, en donde los rubros de mayor impacto son los insumos, equipos y herramientas.

Por tanto, desde el año 2015, se han generado políticas que permitan una dinámica productiva y comercial que ha permitido disminuir la importación del aguacate, abasteciendo la demanda interna, a la par de generar un posicionamiento en el mercado internacional.

Así pues, en el año 2016 se generó un proyecto innovativo en el campo industrial de la producción de aguacate², que buscaba generar un prototipo funcional para el reconocimiento de la madurez del aguacate mediante la implementación de métodos no invasivos en el proceso de selección, usando algoritmos de procesamiento de imágenes relacionado a una base de conocimiento, que pretendía una disminución en los costos de producción de la cosecha.

El proyecto anteriormente citado, tuvo una propuesta que se adecúa a las necesidades del mercado, dado que plantea una herramienta tecnológica que puede ser implementada en el proceso de selección del aguacate Hass, y que en la actualidad se realiza de forma manual, de básculas que de igual manera requieren procesos posteriores de selección, o de máquinas que, por medio de fotografías y análisis con rayos infrarrojos, buscan medir la madurez del aguacate.

² W. E. Choque Valderrama, C. C. Salazar Salazar, “Desarrollo de algoritmo y prototipo móvil para medir el grado de madurez del aguacate Hass mediante procesamiento digital de imágenes”, Universidad Tecnológica de Pereira, 2016.

Sin embargo, el estado alcanzado por el proyecto anteriormente citado tiene unas limitantes visibles que presentan unas oportunidades de mejora resaltables como por ejemplo el hecho de permitir el análisis de un único aguacate, y su nivel de precisión.

Por tal motivo, en este proyecto se propone el uso de una arquitectura de redes neuronales convolucionales, que ha sido previamente utilizada para el reconocimiento de colores de vehículos, y que obtuvo, según sus resultados expuestos, una precisión de 94,47%, valor que representa un alto grado de efectividad en la clasificación y que resulta fundamental para este proyecto.

Así mismo y mediante el uso de herramientas open-source, se crean unos programas que pueden ser ejecutados, para generar archivos de múltiples imágenes para luego ser clasificadas en una única ejecución, por la red neuronal implementada.

DEFINICIÓN DEL PROBLEMA

En la actualidad, Colombia se enfrenta a nuevos contextos internacionales en términos comerciales, en donde una de sus últimas apuestas ha sido la exportación del aguacate en diversas especies³.

Así pues, y con la finalidad de ser mundialmente competitivos, se hace necesaria la investigación sobre nuevas herramientas y tecnologías que permitan, facilitar tareas operativas, que presenten una optimización a los procesos de selección y que reduzcan los costos en la cadena de producción.

Por tanto, se identifica una necesidad a nivel industrial, en donde se pide una propuesta tecnológica que permita la clasificación de la madurez del aguacate, por medio de métodos no invasivos, que además representen el menor impacto económico en cuanto a su desarrollo, instalación y adaptación a la cadena de producción.

³ Ministerio colombiano de agricultura, 2019, “Cadena de aguacate, indicadores e instrumentos”.

JUSTIFICACIÓN

Como una mejora operativa al proceso de clasificación del grado de madurez de aguacates Hass, en empresas de producción y exportación de estos frutos, se hace necesario el desarrollo una herramienta tecnológica capaz de llevar a cabo el proceso de determinación del grado de madurez del aguacate Hass, automatizando dicho proceso.

Es a su vez necesario que la solución propuesta no comprometa en ninguna forma la integridad física del fruto, es decir que utilice métodos no invasivos para el análisis y clasificación de este.

Se busca además que la inversión en la implementación de esta solución no represente costos altos para ninguna compañía, sea grande o pequeña, ya que se busca que sea una herramienta que pueda ser replicada e implementada en sus procesos sin comprometer la liquidez de esta.

Por lo anterior, se plantea el desarrollo de una solución tecnológica que permita el análisis no invasivo del grado de madurez del aguacate Hass, por medio de la toma de fotografías, en una arquitectura que se implementará sobre sobre plataformas open-source, que permitirían una solución que se ajusta a las necesidades expuestas en la industria.

ALCANCE

El proyecto plantea un clasificador del grado de madurez del aguacate haciendo uso de una arquitectura de redes neuronales convolucionales, la cual será plasmada en código fuente, pero no se incluye la generación de un prototipo móvil, ni la implementación de esta de forma física.

OBJETIVOS

OBJETIVO GENENERAL

- Generar una propuesta tecnológica que permita la clasificación de aguacates de acuerdo con su grado de madurez, según el estándar internacional de las normas CODEX STAN 197 – 1995 [23].

OBJETIVOS ESPECIFICOS

- Proponer una arquitectura de red neuronal convolucional que se adapte a las condiciones especiales del problema.
- Proponer un clasificador del grado de madurez del aguacate Hass que funcione en ambientes no controlados.
- Presentar los resultados obtenidos al utilizar la arquitectura de redes neuronales convolucionales propuesta.

HIPÓTESIS

Es posible realizar la clasificación de la madurez del aguacate, utilizando arquitecturas de redes neuronales convolucionales.

METODOLOGÍA

En esta sección se realizará una descripción sobre la metodología de investigación que se llevó a cabo para el desarrollo de este proyecto.

TIPO DE INVESTIGACIÓN

El tipo de investigación desarrollada corresponde a la categoría cuantitativa-experimental, ya que se realizan inferencias a partir de los resultados obtenidos sobre los datos muestrales que se utilizaron como fundamento para este proyecto de investigación.

MÉTODO DE INVESTIGACIÓN

- Escoger una escala de madurez internacional que aplique sobre aguacates Hass.
- Generar un conjunto de imágenes muestrales, que correspondan a aguacates Hass pertenecientes a las categorías planteadas bajo la escala de madurez internacional correspondiente.
- Separar las imágenes de acuerdo con la clasificación establecida.
- Segmentar la totalidad de las imágenes en tres conjuntos, uno de entrenamiento, uno de validación y otro de evaluación, que serán utilizados para generar los resultados finales de clasificación sobre la arquitectura propuesta, usando redes neuronales convolucionales.

- Obtener resultados a partir de la clasificación realizada, utilizando un mismo conjunto de imágenes de evaluación.

TÉCNICAS PARA LA RECOLECCIÓN DE INFORMACIÓN

- Producto de un trabajo de campo se realizó la toma de 275 imágenes⁴ de aguacate Hass.
- Se contó con la ayuda de personas expertas, especializadas en el comercio y exportación de aguacates Hass para realizar la segmentación de los conjuntos de imágenes, en las categorías propuestas para esta investigación (Extra, Categoría I, Categoría II), las cuales constituyen los procesos operativos en las empresas objeto del trabajo de campo.
- El conjunto de imágenes fue obtenido en condiciones naturales lumínicas, esto quiere decir, que están expuestas a ruidos y variaciones sobre las mismas, y por tanto se clasifican como imágenes en ambientes de ruido no controlados.
- Las imágenes fueron obtenidas con una cámara tipo Evil Samsung NX 300, con un lente fijo de 45 mm, apertura de diafragma de f. 1.8

⁴ Repositorio de imágenes : <https://github.com/alejomoreno07/AvocadoKB>

ESTUDIO DE CAMPO

Se realizó un estudio de campo en tres empresas de exportación de aguacate Hass, las cuales son Cartama⁵, FLP⁶ y Hass Diamond⁷.

Cartama corresponde a la empresa exportadora más grande de Colombia, con una trayectoria de más de 15 años, que cultiva y comercializa Aguacate Hass, y que cuenta con unidades productivas en diferentes zonas de Colombia, sin embargo, tiene su planta de embalaje en la ciudad de Pereira, Risaralda.

Así mismo, FLP (Fresh fruits & vegetables) es una empresa pionera en la exportación de frutas exóticas, que cuenta con presencia en Ecuador, Perú y Colombia.

Por último, Hass Diamond, es una empresa exportadora de aguacates Hass, la cual surge inicialmente como una empresa que presenta alternativas de ofertas y comercialización para productores de aguacates, y que luego de más de 10 años de experiencias en el sector, surge en el 2016 como una empresa exportadora del mismo.

En la visita de campo se encuentra que, FLP y Hass Diamond realizan un proceso mecánico de clasificación mediante el uso de básculas, que identifica, mediante el peso del fruto, si este se encuentra en las condiciones requeridas bajo los estándares de exportación, o si se deja para la comercialización nacional.

⁵ Página web de la empresa Cartama: <https://cartama.com.co/>

⁶ Página web de la empresa FLP : <http://flp.co/>

⁷ Página web de la empresa Hass Diamond: <http://www.hassdiamond.co/en/>



Ilustración 1: báscula de clasificación de aguacates Hass, según su peso. Fuente: autoría propia.

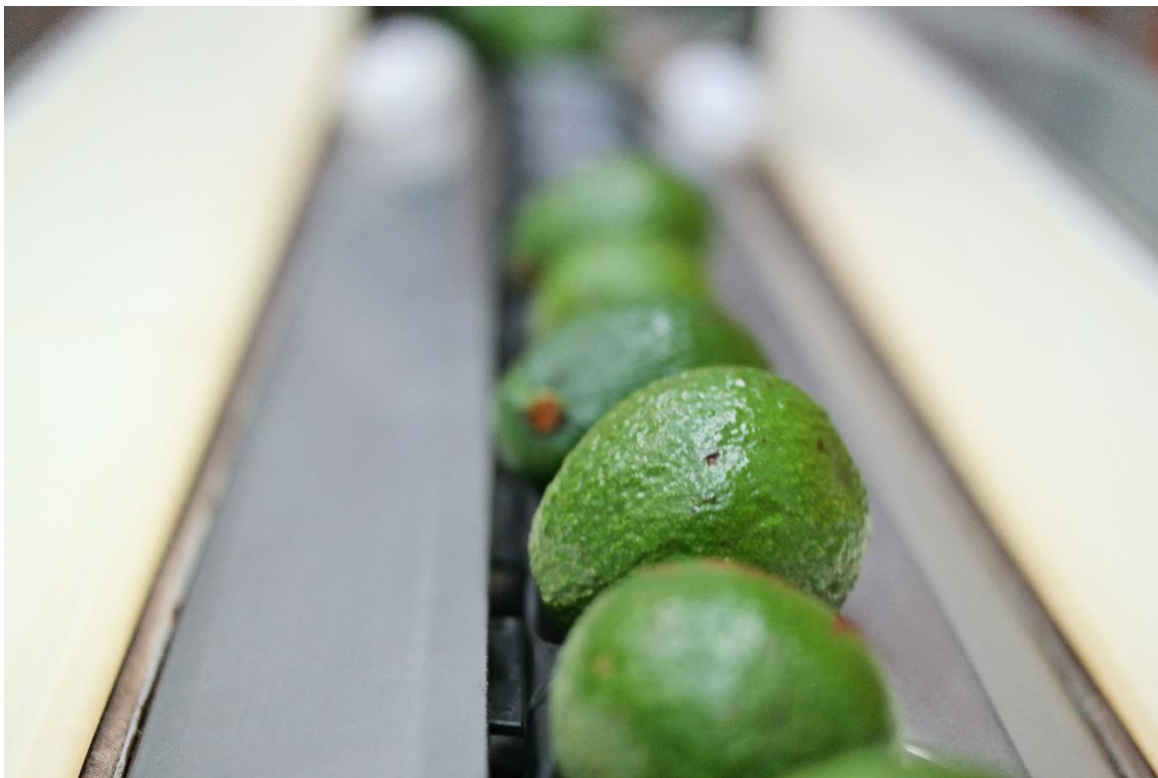


Ilustración 2 : Transporte y depósito de los aguacates en su correspondiente calibre, Fuente: autoría propia.



Ilustración 3: Transporte y depósito de los aguacates en su correspondiente calibre, Fuente: autoría propia.

Por último, y dado que para la correcta clasificación del fruto se requieren además del peso otros factores físicos como el color, la textura, la cicatrización, entre otros, se realiza un proceso manual de verificación manual, el cual corresponde a un esfuerzo humano, que es susceptible a fallas y que puede además impactar sobre el producto final de la empresa.

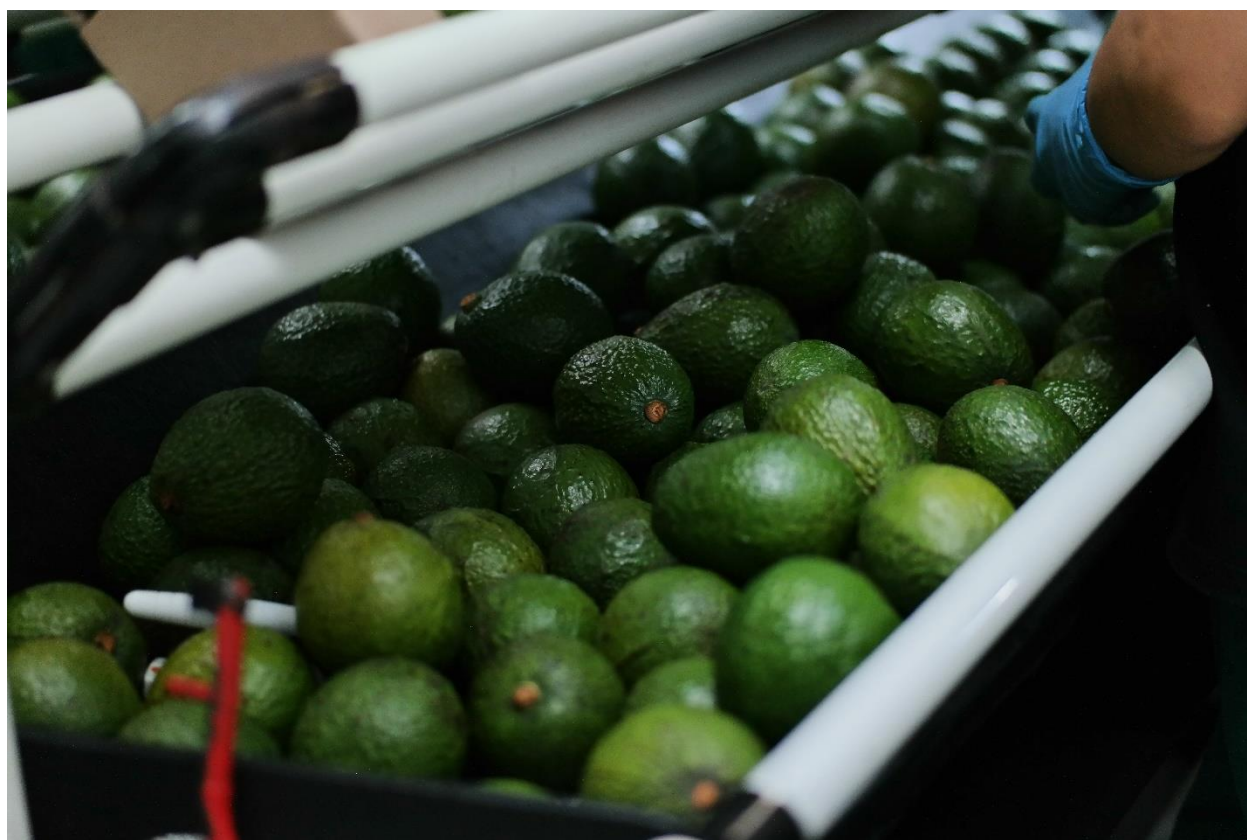


Ilustración 4: proceso de selección manual del fruto. Fuente: autoría propia.

MARCO DE REFERENCIA

- Overfitting: hace referencia a la dificultad de generalizar los datos obtenidos a partir del entrenamiento, lo que hace incapaz de inferir una correcta clasificación de nuevos datos.
- Ciclos de entrenamiento: se refiere a una iteración completa sobre el conjunto completo de entrenamiento utilizando el algoritmo de aprendizaje.
- Max pooling: es un proceso utilizado para reducir la dimensionalidad de un conjunto de datos, agrupándolos y utilizando una maximización de los valores de dicho grupo como representación de este.
- Grados de madurez del aguacate Hass: según la norma CODEX-STAN-197-1995 [23], se establecen las siguientes categorías de clasificación:
 - a. Extra: son aquellos que corresponden a la categoría de calidad superior, los cuales no deberán tener defecto alguno, exceptuando defectos superficiales leves en la cáscara, que no afecten el aspecto general del producto.
 - b. Categoría I: son aquellos que son de buena calidad, sin embargo, pueden permitirse defectos leves con respecto a la forma y coloración del mismo, así mismo como defectos leves en la cascara y quemaduras producidas por el sol, sin afectar de ninguna manera la pulpa del fruto.
 - c. Categoría II: son aquellos que no pueden clasificarse en las dos anteriores categorías, ya que poseen ciertos defectos de forma y coloración, o defectos de la cáscara, pero que conservan todavía sus características esenciales en lo que respecta a su calidad y estado de conservación.

MARCO TEÓRICO

En este capítulo se presentarán las teorías que se relacionan con el contexto del problema de investigación, teniendo como fin último, ofrecer un sustento al estudio que se está desarrollando en el presente proyecto.

APRENDIZAJE DE MÁQUINAS

El aprendizaje de máquinas⁸ constituye un subcampo en las ciencias computacionales, que hace referencia a la detección automática de patrones significativos en conjuntos de datos, y a las aplicaciones computacionales que proveen mecanismos de adaptación que permiten o facilitan un comportamiento inteligente en entornos complejos, inciertos y cambiantes.

Básicamente, el aprendizaje de máquinas se basa en utilizar las características presentes en el conjunto de datos para construir los modelos que logren realizar unas determinadas tareas.

De esta manera, dichas características definen un lenguaje en el que se pueden describir los objetos relevantes en el dominio.

⁸ Deep Learning, Yann LeCun, Yoshua Bengio, Geoffrey Hinton, 2015, Nature Research Journal

Así mismo, las tareas que se quieren llevar a cabo son en realidad una representación abstracta de un problema que se quiere resolver, con respecto a las características, anteriormente definidas.

Y como último elemento constitutivo, se tienen el modelo, que en sí mismo representa el resultado de un algoritmo de aprendizaje de máquina, aplicado sobre un conjunto de datos de entrenamiento.

RED NEURONAL

Es un conjunto interconectado de simples elementos de procesamiento, unidades o nodos, cuya funcionalidad se basa en la de una neurona animal. La habilidad de procesamiento está contenida en la fuerza de conexión entre las unidades, o el peso de sus conexiones, obtenido por un proceso de adaptación o aprendizaje a un conjunto de patrones de entrenamiento⁹.

De esta manera, las redes neuronales están compuestas por un conjunto de neuronas interconectadas entre si mediante enlaces.

Cada uno de los elementos de la red neuronal, toma como entrada las salidas de la capa antecesora, multiplicando cada una de ellas por una cantidad denominada peso, que funciona en este caso como el grado del estímulo.

⁹ An introduction to neural networks, Kevin Gurney, 1997, University College London

La neurona calcula su salida con respecto a una función de activación, la cual es una función matemática, que se adapta a las características propias del problema y la arquitectura.

A su vez, la salida de una neurona hace parte del estímulo que recibe la siguiente capa y así sucesivamente.

En el paradigma de las redes neuronales, se tienen varias fases que buscan refinar el peso de los estímulos, con el fin de alcanzar un punto que permita obtener los mejores resultados, de acuerdo con el problema que se plantee.

De esta manera, la interacciones entre estos simples elementos, generan un modelo que es capaz por medio del entrenamiento, de realizar tareas complejas como distintas clasificaciones o detecciones.

RED NEURONAL CONVOLUCIONAL

Las redes neuronales convolucionales¹⁰ son también un conjunto interconectado de elementos de procesamiento, que puede tomar una imagen y asignar importancia a

¹⁰ An introduction to convolutional neural networks, 2015, Keiron Teilo O'Shea Ryan Nash, Springer.

ciertos aspectos, características, clases u objetos en la imagen, siendo capaz de diferenciarlos entre ellos.

Uno de los aspectos que diferencian a las redes neuronales convolucionales con sus predecesoras, las redes neuronales, es el preprocesamiento requerido, el cuál para este caso resulta ser mucho menor a los que utilizan los algoritmos de clasificación de estas últimas.

Su arquitectura es un análogo a los patrones de interconexión de las neuronas en el cerebro humano, inspirándose en la organización del Cortex Visual.

De esta manera, en una red neuronal convolucional, neuronas individuales responden a estímulos sólo en una restringida región del campo visual, conocida como campo receptivo. Un conjunto de dichos campos, se sobreponen para cubrir completamente el campo visual.

DETECCIÓN DE OBJETOS

El problema de detección de objetos¹¹ envuelve dos tareas fundamentales, que han sido objeto de diversos métodos de solución en el transcurso de los años, como parte del campo de visión por computadora. Estos dos problemas son la clasificación y la localización de objetos.

¹¹ Object detection with Deep learning: A review, 2019, Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, Xindong Wu, IEEE Transactions on neural networks and learning systems.

Para este problema, tomando como entrada una imagen, la salida sería un objeto (generalmente un rectángulo) que enmarque a cada uno de los distintos objetos detectados, es decir, un objeto que nos de la clasificación y localización encontrada.

Generalmente, las soluciones a este problema tienen diversas aplicaciones, como detección de rostros, carros, etc. Sin embargo, una versión más generalizada de este problema plantea básicamente la detección de múltiples objetos como parte de sistemas de vigilancia.

MÁQUINAS DE SOPORTE VECTORIAL

Las máquinas de soporte vectorial¹² son una pieza y herramienta fundamental en la minería de datos, basándose en técnicas de aprendizaje estadístico y métodos de optimización para solucionar problemas clásicos en el campo de aprendizaje de máquinas, al ser utilizadas entre muchas otras cosas, para problemas de conjuntos finitos de entrenamiento, para superar overfitting, entre otros.

Entre sus muchas características se destacan el poseer unas buenas representaciones matemáticas, explicaciones geométricas, y unas buenas capacidades de generalización, que son el reflejo de su uso ampliamente aceptado en el medio.

¹² “Support Vector Machines Optimization Based Theory, Algorithms, and Extensions”, 2013, Naiyang Deng, Yingjie Tian, Chunhua Zhang, Taylor & Francis Group.

Las máquinas de soporte vectorial son entonces una herramienta que se usa en problemas de clasificación y regresión por medio de un aprendizaje supervisado en donde se toma un conjunto de datos de entrenamiento como puntos pertenecientes a un vector de la dimensión de dicho conjunto, y lo que se busca es encontrar un hiperplano que permita separar de manera óptima los elementos pertenecientes a cada clase.

ESTADO DEL ARTE

En este capítulo se presentarán algunos de los estudios desarrollados a lo largo de los años con el fin de analizar distintos planteamientos metodológicos, arquitecturales y algorítmicos, y la forma en cómo estos planteamientos han permitido abordar problemas similares desde las diversas áreas, creando una base importante para el desarrollo y la dirección del proyecto.

Desarrollo de algoritmo y prototipo móvil para medir el grado de madurez del aguacate Hass mediante procesamiento digital de imágenes.

Autores: William Enrique Choque Valderrama y Cristian Camilo Salazar Salazar

Año: 2016

Descripción:

En este texto correspondiente a una tesis de pregrado, en el año 2016, se aborda el estudio de la correlación existente entre el color de los aguacates del tipo Hass con respecto a la maduración de este, encontrando una aplicación interesante a las redes neuronales, generando un prototipo en una plataforma móvil capaz de generar una clasificación del grado de madurez del aguacate.

A su vez presentan los resultados correspondientes a la clasificación de aguacates en ambientes controlados, (es decir, en donde se eliminan ruidos en las imágenes) y se presentan los resultados obtenidos.

Aportes a la tesis: el prototipo presentado como solución tecnológica en el contexto industrial agrícola es la base para el desarrollo del presente trabajo de investigación, presentando una visión del objetivo último a desarrollar, el cual consiste en proporcionar una herramienta tecnológica ante la industria local, como apoyo a su labor productiva, la cual fue desarrollada como respuesta a una necesidad expresada por parte de empresas exportadoras de aguacates Hass como parte de un proceso de análisis operativo, sobre su proceso de selección.

Así mismo, y mediante un análisis más detallado, se encuentran opciones de mejora que se pretenden incluir en el curso de este proyecto.

Ripeness Classification of Bananas Using an Artificial Neural Network

Autores: Fatma M. A. Mazen, Ahmed A. Nashat

Año: 2019

Descripción:

El artículo presenta un Sistema de Visión por computadora, utilizando máquinas de soporte vectorial (SVM) y algoritmos como KNN, árboles de decisión y clasificadores de discriminantes, el cuál logra una efectividad de un 97.75%.

En su trabajo, se explora el proceso de selección de bananos, en el cuál se describen además cómo el proceso de selección manual puede afectar considerablemente la calidad del fruto y además puede entorpecer su proceso de comercialización.

A su vez, se detalla un proceso metodológico, en el cual se inicia posterior a la adquisición de las imágenes, en la que se realiza un preprocesamiento sobre las mismas, extrayendo las frutas del fondo de la imagen, para evitar ruidos inducidos por el sistema, y luego extraer a su vez, las características principales de la fruta y de esta manera realizar su correcta clasificación.

De igual manera, se detalla cómo pudieron lograr estos niveles de precisión con una base de conocimientos de 300 frutas, en distintas etapas de maduración.

Por último, presentan una serie de ecuaciones con el fin de determinar el nivel de madurez, y a su vez el árbol de decisión que se planteó.

En las conclusiones a partir de los resultados obtenidos se expone una gran efectividad presentada al utilizar la arquitectura propuesta, en donde algunas clases fueron siempre clasificadas correctamente.

Aportes a la tesis: resulta relevante e innovativo para la industria agrícola, y además constituye un gran aporte dado que se propone un proyecto que intenta resolver, al igual que este trabajo, un problema sobre la selección de frutos de acuerdo con características físicas del mismo.

Esta investigación, además, es la inspiración sobre trabajos futuros para la actual investigación, en donde se propondrán arquitecturas de inteligencia artificial distintas a las redes neuronales, como, por ejemplo, las máquinas de soporte vectorial.

Image processing applied to classification of avocado variety Hass (Persea Americana Mill.) during the ripening process

Autores: Israel Arzate-Vázquez, M.J. Perea-Flores, Georgina Calderón-Dominguez, Jorge Chanona-Perez

Año: 2011

Descripción:

Como parte de un proceso de análisis de maduración de los aguacates Hass, se utilizan métodos de procesamiento de imágenes, en donde se toma un conjunto de ellos para llevar a cabo un seguimiento sobre los cambios en las características físicas, implementando un sistema de visión por computadora, extrayendo parámetros como el color y la textura.

Como conclusión de la investigación, se encuentra con que las características de las imágenes obtenidas mediante procesamiento de imágenes brindan una buena descripción del proceso de madures de los aguacates Hass, encontrando que en especial el color y la textura de estos, tienen una alta correlación con su calidad.

Aportes a la tesis: fundamenta, por medio de la experimentación, las hipótesis sobre la correlación existente entre el color y la textura y la calidad del aguacate, siendo el color el principal parámetro para dar un veredicto sobre su madurez.

Vehicle Color Classification Under Different Lighting Conditions Through Color Correction

Autores: Jun-Wei Hsieh, Li-Chih Chen y Salah Alghyaline

Año: 2014

Descripción:

Se presenta una propuesta ante el problema de clasificación de colores de vehículos, bajo diferentes condiciones de luminosidad, utilizando corrección de colores.

Se exploran además, los distintos ruidos a los que pueden ser expuestas las imágenes, y las distintas problemáticas que representan al proceso de clasificación.

Se habla entonces, de cómo existen errores comunes al momento de la clasificación inducidos, en donde, por ejemplo, se etiquetan dos diferentes colores como uno solo, o cómo las condiciones de luz u oscuridad pueden entorpecer el proceso.

Sin embargo, la investigación propone una serie de metodologías a utilizar para remediar estos ruidos inducidos en ambientes no controlados.

Por tanto, se utilizan métodos de preprocesamiento, para eliminar pixeles que podrían corresponder a ruidos, ya sea por saturación de luminosidad o por el contrario por sombras, permitiendo condensar los colores más representativos en la imagen, permitiendo así una clasificación más efectiva.

Aportes a la tesis: este artículo en particular explora un campo muy importante que se plantea como un trabajo a futuro para este proyecto, y es el de generar una mejora adaptativa sobre ruidos en las imágenes.

Vehicle Color Recognition using Convolutional Neuronal Network

Autores: Reza Fuad Rachmadi, Ketut Eddy Purnama

Año: 2018

Descripción:

El problema que expone el artículo es una variación a un planteamiento clásico dentro del campo de procesamiento de imágenes, el cual consiste en el reconocimiento de objetos, para el que existen variadas arquitecturas y metodologías, en el campo del Deep Learning, que han sido objeto de análisis en la literatura.

Sin embargo, su variación consiste al no sólo intentar reconocer el objeto, sino además identificar su color.

En el año 2018, la investigación siendo sucesora de varios planteamientos en el campo, propone la implementación de una arquitectura de red neuronal convolucional, generando además un análisis sobre las funciones de activación y las metodologías que se utilizaron en su experimentación.

Su uso no es convencional, ya que, como se expresa en el artículo, generalmente las arquitecturas de redes neuronales convolucionales se han utilizado para clasificar

objetos basados en su figura, pero logran demostrar que también puede ser utilizada en un modo efectivo para la clasificación de objetos según su distribución de colores, obteniendo mejores resultados que aquellos usados por sus predecesores.

Aportes a la tesis: el artículo presenta argumentos sólidos que no solo reafirman la Hipótesis central de este proyecto, presentando datos reales sobre mejoras, sino que, además, permite por medio de su arquitectura, y código libre, implementar su solución al problema a resolver.

Colour Recognition in Images Using Neural Networks

Autores: R. Viegnessawar, Ms. V. Prema

Año: 2016

Descripción:

Se presenta una arquitectura de redes neuronales interesante, en cuanto propone, con el fin del reconocimiento de colores, la creación de capas de redes neuronales intermedias, especializadas en el reconocimiento particular de ciertos colores, y de esa manera, creando una arquitectura jerárquica, poder al final encontrar la clasificación sobre el color de la imagen.

Sin embargo, y pese al uso de esta arquitectura, se deja entrever el uso de categorías de colores predefinidas, es decir una categorización de acuerdo con sus valores RGB.

Aportes a la tesis: el artículo presenta una idea innovativa sobre una arquitectura de redes neuronales, para el reconocimiento de colores, que, si bien no usa redes

neuronales convolucionales, es una nueva postura que complementa el contexto teórico buscado, al momento de comparar y decidir sobre qué arquitectura se usará como solución tecnológica para el presente proyecto.

Assessment of Ripening Degree of Avocado by Electrical Impedance Spectroscopy and Support Vector Machine

Autores: Monzurul Islam, Khan Wahid, Anh Dinh

Año: 2018

Descripción:

Esta investigación explora el problema de clasificación de la etapa de madurez del aguacate, utilizando métodos invasivos y no destructivos, para determinar su correcta clasificación.

En el estudio se menciona cómo se generan mediciones sobre las impedancias eléctricas en los frutos, debido a que estos tienen una estructura biológica que se compone de tejidos con células que están suspendidas en fluidos extracelulares, que a su vez proveen un camino para aplicar una corriente alterna sobre dichos tejidos, los cuales generan una bioimpedancia eléctrica como respuesta a esta exposición.

Al final de estas mediciones, y por medio de máquinas de soporte vectorial, se realiza la clasificación del grado de madurez del aguacate en cuestión sobre las categorías planteadas.

Aportes a la tesis: este trabajo muestra un enfoque distinto para el mismo problema en este proyecto, y resulta interesante dado que resalta cómo por medio de propiedades químicas y físicas, puede determinarse también el grado de madurez del aguacate. También presenta una de las justificaciones de este proyecto, en cuanto la realización de este proceso de clasificación, por su naturaleza invasiva, puede poner en riesgo la integridad del fruto, que en un contexto industrial de exportación puede conllevar a riesgos muy elevados.

Skin Color Detection Model Using Neural Networks and its Performance Evaluation

Autores: K.K. Bhoyar, O.G. Kakde

Año: 2010

Descripción:

Este artículo presenta una propuesta arquitectónica, por medio del uso de redes neuronales, como solución al reconocimiento de color de piel en imágenes.

Es relevante, en cuánto se aborda el problema de reconocimiento de colores a la par de servir como base para problemas más complejos como el reconocimiento de caras y personas.

El artículo propone la clasificación de las imágenes, mediante la división de las imágenes en regiones de piel y no-piel, en donde una arquitectura de red neuronal de tres capas se encarga de la clasificación del color.

Aportes a la tesis: este trabajo presenta una oportunidad para futuros trabajos sobre el presente proyecto, dado que la segmentación de regiones en donde el fruto está presente podría reducir el ruido en las imágenes y a su vez incrementar el grado de precisión de la clasificación.

Olive Ripening Phase Estimation based on Neural Networks

Autores: Marco Mora, Jorge Aliaga, Claudio Fredes

Año: 2017

Descripción:

Se explora el campo de la clasificación del grado de madurez de olivas, por medio del reconocimiento del color de estas utilizando una arquitectura de redes neuronales.

En su investigación, se genera un primer paso en donde se crea una base de conocimiento de colores, por medio de la clasificación en categorías de forma manual. Luego se procede a entrenar la red neuronal con la base de conocimiento resultante del proceso anterior, y por último se realiza la clasificación por medio del uso de la red neuronal.

Además, se expone cómo se utilizaron medidas para reducir el ruido en las imágenes, tales como el uso de soportes al momento de tomar las fotos o el preprocesamiento de imágenes, con el fin de reducir el impacto en la clasificación.

Aportes a la tesis: al presentarse una metodología procedimental válida, se plantea su uso para el desarrollo de este proyecto, y además, expone nuevos campos en la industria agrícola a los cuales podría aplicarse la arquitectura que se utilizará en este proyecto.

Color names learning using convolutional neural networks

Autores: Yuhang Wang, Jing Liu, Jinqiao Wang, Yong Li, Hanqing Lu

Año: 2015

Descripción:

Este artículo propone el reconocimiento de colores por medio una arquitectura de redes neuronales convolucionales de dos etapas, en donde la primera etapa corresponde al reconocimiento del color de pequeñas secciones en una imagen, en donde además tiene etiquetas, que representan ruido, y que son superadas dado que, en ambas etapas, las redes convolucionales son a su vez auto supervisadas.

En la segunda etapa, se infiere el color que mejor representa la sección a clasificar, y, por tanto, se puede pensar estas dos etapas, como primero, la extracción de las características físicas principales, y la segunda, como la inferencia del color de este.

Se hace uso además, de un histograma de colores basado en clusters y a su vez se modifica la función de pérdida “softmax” para obtener mejores resultados.

Aportes a la tesis: este artículo abre la puerta a un posible trabajo futuro, en donde se utilizan dos etapas de procesamiento para generar un mayor grado de precisión en la clasificación de colores.

Deep indicator for fine-grained classification of banana's ripening stages

Autores: Yan Zhang, Jian Lian, Mingqu Fan, Yuanjie Zheng

Año: 2018

Descripción:

Este trabajo corresponde a la primera investigación, en donde se hace uso de una arquitectura de redes neuronales convolucionales para el reconocimiento de la etapa de maduración del banano.

Para tal fin, se presenta un desarrollo que ha conseguido una efectividad en la clasificación, por medio de imágenes, del 94,4%, utilizando una base de conocimiento de 17312 imágenes.

En su arquitectura, se propone una estructura jerárquica multicapa, que además presenta unos campos receptores locales, pesos compartidos, y unas capas de agrupación espacial.

En este sentido, la red neuronal convolucional captura segmentos específicos de la imagen, extrayendo sus características principales, en vez de procesar la totalidad de la imagen.

Así mismo, el sentido de los pesos compartidos para distintas neuronas de una misma capa se da cuando se realiza la evaluación de un mismo filtro sobre los segmentos de la imagen que se está clasificando.

Por último, se aplica la agrupación espacial, que tiene como finalidad reducir la dimensionalidad de las respuestas de la convolución y mejorar la invarianza traslacional en el modelo, tomando la mayor respuesta por cada segmento sobre todas las respuestas en la agrupación.

Aportes a la tesis: este trabajo en particular presenta una línea muy similar a lo planteado en este proyecto, al proponer una metodología de clasificación no invasiva para la clasificación de madurez del banano, y en donde se tiene un alto grado de precisión. Este trabajo además explora nuevas direcciones que se están tomando en la innovación y desarrollo industrial para productos agrícolas, y fue analizado entre las arquitecturas a usarse para este proyecto.

Automatic Recognition of Ripening Tomatoes by Combining Multi-Feature Fusion with a Bi-Layer Classification Strategy for Harvesting Robots

Autores: Jingui Wu, Baohua Zhang, Jun Zhou, Yingjun Xiong, Baoxing Gu, Xiaolong Yang.

Año: 2019

Descripción:

Este trabajo propone un procesamiento de imágenes y una clasificación posterior sobre la madurez de los tomates, de acuerdo con un algoritmo desarrollado para tal fin.

En su primera etapa, posterior a la adquisición de las imágenes, en las que se resalta que se toman medidas para evitar ruidos como exceso de luminosidad, se realiza un procesamiento en el que se extraen las regiones que contienen tomates, y luego, dicha región será objeto de análisis por el algoritmo para su clasificación.

En el algoritmo, las características extraídas de las imágenes se usan para generar el vector de pesos que será el elemento principal en la maquina vectorial de relevancia, que es precisamente, la tecnología usada para crear la correcta clasificación de la madurez de los tomates.

Para finalizar, se establece un porcentaje de precisión del 94.90% y un tiempo promedio de ejecución por clasificación de 2.94s

Aportes para la tesis: este trabajo es una posible solución al problema de clasificación por colores, presentando a su vez unos resultados sobresalientes en contraste a las otras soluciones en la industria, además su tiempo de ejecución es relativamente corto y podría ser implementado en procesos agrícolas sin poner en riesgo la integridad del producto.

Prediction of banana quality indices from color features using support vector regression

Autores: Alireza Sanaeifar, Adel Bakhshipour, Miguel de la Guardia

Año: 2015

Descripción:

Este trabajo consiste en la experimentación sobre 15 bananos que fueron llevados a través de las distintas etapas de maduración en ambientes controlados de temperatura y humedad.

A su vez, se crea un sistema de visión por computadora, el cuál consiste en dos partes, una para la adquisición de imágenes y la otra que se encargará de la clasificación de las imágenes obtenidas.

El componente de adquisición de imágenes tiene como función principal disminuir ruidos en las imágenes tales como exceso de brillo u opacidad que puedan obstruir con el proceso.

El segundo componente, se basa en la segmentación de la imagen en regiones de interés, es decir regiones que contengan información sobre el fruto, separándolas de aquellas que contienen, por ejemplo, el fondo de la imagen.

En medio de esta segmentación, se obtiene una imagen binaria de la imagen, en donde se separa la región de interés, del fondo, y luego es aplicado un operador lógico AND con los colores originales de la imagen para no perder las características relevantes de esta.

Al final, y por medio de una máquina de soporte vectorial, se realiza la clasificación de los resultados obtenidos en las categorías de madurez correspondientes.

El trabajo de investigación finaliza presentando una precisión del 92%, y expresa a su vez entre sus conclusiones, cómo este método podría ser implementado en la industria ya que posee un nivel de precisión alto y tiene en promedio tiempos de procesamiento menores a los que podrían ser encontrados bajo redes neuronales convolucionales.

Aportes a la tesis: este trabajo en específico abre la puerta a nuevos estudios comparativos sobre el aguacate, y su posible implementación podría generar nuevos intentos al presentar métodos no invasivos que busquen aumentar la eficiencia y la eficacia para procesos tan críticos como la clasificación automatizada de frutos.

Fruit ripeness estimation for avocado using thermal imaging

Autores: Sathish k. Gurupatham, Nick Jacob, Kevin Van Der Horn, Erhan Ilksoy, Fahad Fahad

Año: 2018

Descripción:

Este trabajo explora la toma de imágenes térmicas para la determinación del grado de madurez del aguacate tipo Hall, dado que dichas imágenes térmicas pueden ser utilizados en frutos que no cambian de verde a amarillo en sus etapas de maduración, según lo expresado.

De esta manera, se plantea además que existe un grado de correlación entre la temperatura y las características de la etapa de maduración en la que se encuentra el fruto.

A su vez, fue el experimento fue desarrollado en un ambiente controlado de temperatura, en el cuál se realizaba el seguimiento térmico de los frutos en medio de los cambios de las etapas de maduración, encontrando que, con el paso del tiempo, y por ende el paso de una etapa de maduración a otra, se aumenta la temperatura medida en este proceso.

Aportes a la tesis: este proyecto presenta una posibilidad para generar una clasificación de imágenes térmicas usando redes neuronales convolucionales, como un posible trabajo futuro, y a su vez como una alternativa a los métodos no invasivos que son el fin último de la industria local.

Apple Ripeness Estimation using Artificial Neural Network

Autores: Raja Hamza, Mohamed Chtourou

Año: 2018

Descripción:

Este artículo presenta un modelo de redes neuronales para la determinación de la madurez de las manzanas, utilizando una arquitectura basada primero en el preprocesamiento de las imágenes, en donde se redimensionan y luego son extraídas las regiones de interés de las imágenes, es decir, se extrae el fruto de las imágenes.

Posterior a ello, de este resultado, son extraídas las características más relevantes, mediante los colores presentes en la región de interés, para luego, y a partir de ellas, ser clasificadas en la etapa de madurez correspondiente.

El artículo además presenta una precisión obtenida del 98.33% con respecto a los datos de validación que se utilizaron.

Aportes a la tesis: este artículo presenta un posible proceso metodológico para el mejoramiento de la precisión en modelos que usan redes neuronales, sin tener que recurrir a modelos como redes neuronales convolucionales o máquinas de soporte vectorial, siendo un posible trabajo futuro del proyecto de Salazar & Choque, que podría a su vez cumplir con la meta de ser liviano para entornos de plataformas móviles.

Tomato classification according to organoleptic maturity (coloration) using machine learning algorithms K-NN, MPL, and K-Means Clustering

Autores: Wolfgang D. Niño Pacheco, Fabian R. Jiménez López

Año: 2019

Descripción:

En este trabajo se explora el problema de clasificación de tomates utilizando como marco de referencia su color, y haciendo uso de distintos modelos y algoritmos para la realización de la tarea de clasificación, antes mencionada.

En su investigación, Niño Pacheco & Jiménez López describen las diversas etapas de procesamiento de las imágenes, en donde se genera un procesamiento sobre las imágenes, retirando el fondo, y generando una delimitación de las regiones de interés, de las que serán extraídas sus características principales para posteriormente pasar a una etapa de análisis.

Se realiza, además, una normalización de los valores RGB en la región de interés, en donde el histograma generado a partir de sus valores entre 0 y 255, son transformados a valores entre 0 y 1.

Posterior a la obtención de los valores extraídos de las regiones, se pasa a una etapa de clasificación en donde se utilizan los algoritmos de los K vecinos más cercanos (K-NN), agrupación por K medias (K-means clustering) y una red neuronal de tipo perceptrón multicapa.

Según los resultados presentados, K-NN y la red neuronal multicapa presentan precisiones por encima del 90% para cada clase, sin embargo, K-means clustering presenta precisiones alrededor de 80%.

Aportes a la tesis: esta investigación presenta un estudio comparativo entre diversos algoritmos encaminados a la clasificación de imágenes de frutos, teniendo como referencia su color, y además, presenta a las redes neuronales como un modelo apropiado para cumplir con esta tarea, argumento que reafirma la idoneidad de la implantación de una como solución tecnológica en este contexto.

DESCRIPCIÓN DE LOS RESULTADOS Y LA ARQUITECTURA

En este apartado se describirá la arquitectura propuesta e implementada, así como los resultados obtenidos en el proceso de entrenamiento y validación de los resultados sobre una muestra de 275 imágenes¹³ correspondientes a aguacates tipo Hass, en 3 categorías de madurez comercial (exportación, nacionales y maduros), las cuales fueron obtenidas por medio de una visita de campo a la empresa Cartama, la cual corresponde a una de las mayores productoras locales de dicho fruto en la región.

DESCRIPCIÓN DE LA ARQUITECTURA PROPUESTA

Producto al estudio realizado del estado del arte, se llegó a conclusión de utilizar la arquitectura de redes neuronales propuesta por Reza Fuad Rachmadi & I Ketut Eddy Purnama en su artículo titulado “Vehicle Color Recognition using Convolutional Neuronal Network”, en donde se presenta una red convolucional multicapa, especializada en el reconocimiento de colores, aplicado para el problema de dicho reconocimiento en carros.

Para el problema particular, y dada la naturaleza del problema, era necesario reconocer el grado de madurez del aguacate por el color de este, no necesariamente realizar un proceso de segmentación sobre este, y por tanto la propuesta anteriormente citada cumple a cabalidad estas necesidades.

¹³ Repositorio de imágenes en : <https://github.com/alejomoreno07/AvocadoKB>

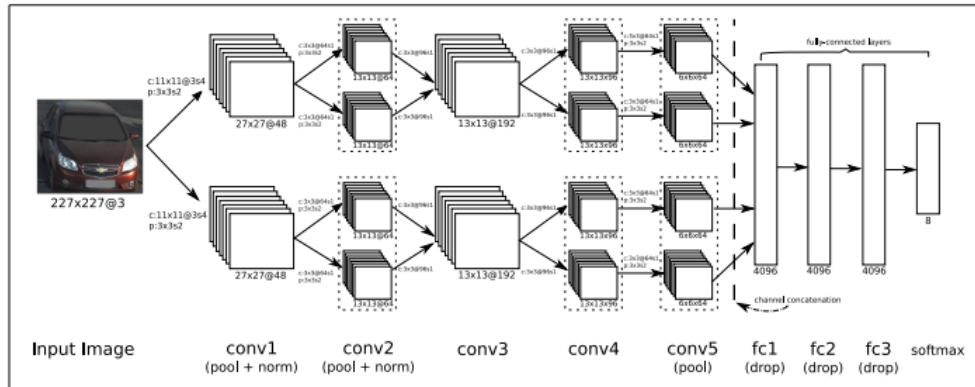


Ilustración 5: arquitectura de red neuronal. Obtenida de “Vehicle Color Recognition using Convolutional Neural Network”

La arquitectura consta entonces, de 8 capas, en dos ramas idénticas, donde tres de ellas realizan un max pooling y luego una normalización, posterior a la realización de la convolución; 2 de ellas sólo realizan una convolución y las restantes unen las dos ramas existentes por medio de capas de neuronas completamente conectadas que unifican los resultados obtenidos.

Fue utilizado entonces, el código fuente que se encuentra en su repositorio, y fue adaptado, conservando el lenguaje de programación Python, emulando su ejecución en Jupyter, que es una herramienta libre que permite la ejecución e instalación de librerías tales como tensorflow, keras, etc.

Adicionalmente a la descripción de la red neuronal convolucional, se adicionó un script que permite de forma fácil, generar conjuntos de datos de entrenamiento y validación, para el entrenamiento de la red neuronal convolucional, así mismo como otro para generar un conjunto de imágenes que serán sometidas a la clasificación del este.

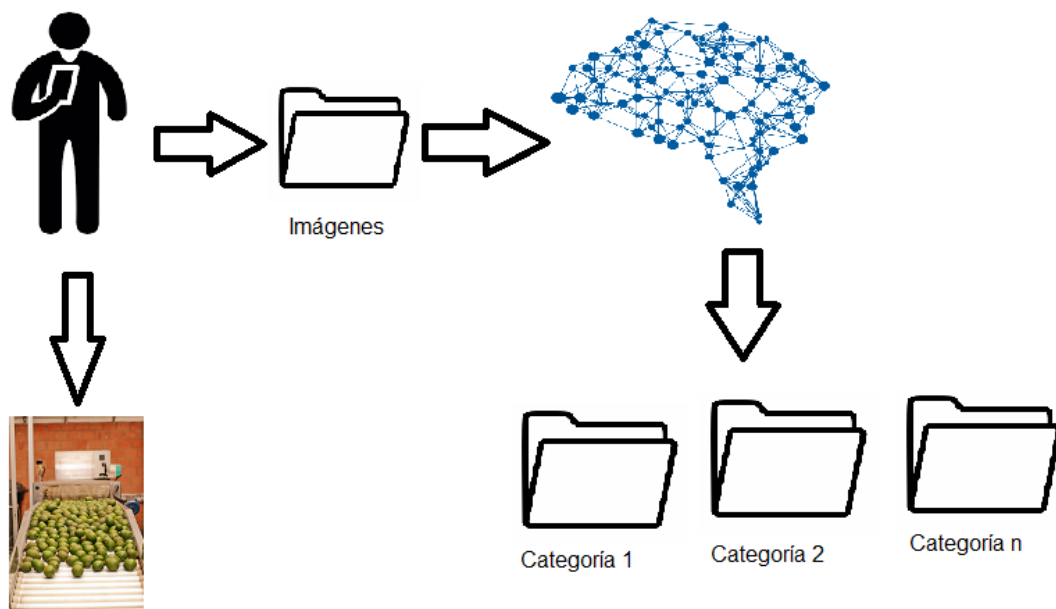


Ilustración 6: proceso de clasificación. Imagen propia.

En la ilustración anterior, se encuentra el proceso propuesto para realizar la clasificación del grado de madurez del aguacate, en donde primero serán tomadas fotografías de aguacates Hass individuales, en donde se almacenarán todas las imágenes en una sola carpeta y luego serán clasificadas al mismo tiempo, generando los resultados de la clasificación, a partir de la base de conocimiento construida previamente, y además, de las categorías planteadas.

A continuación, se realizará una descripción de los bloques de código constitutivos del clasificador.

Se aclara que la disposición inicial de los archivos debe realizarse de acuerdo con las categorías que se utilicen, es decir, las imágenes pertenecientes a la base de conocimientos deben de estar separadas en carpetas que lleven el nombre de dichas categorías, en el arreglo “CATEGORIES”.

La ruta local en donde deben de ser almacenadas las imágenes se debe establecer en la variable “DATADIR”.

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
import os
```

```
import cv2
```

```
from sklearn.preprocessing import label_binarize
```

```
DATADIR = "C:/Users/alejo/proyecto/avocados/"
```

```
CATEGORIES = ["exportation","matures","nationals"]
```

```
IMG_SIZE = 227
```

```
training_data = []
```

```
def create_training_data():
```

```
    for category in CATEGORIES:
```

```
        path = os.path.join(DATADIR,category)
```

```
        class_num = CATEGORIES.index(category)
```

```
        for img in os.listdir(path):
```

```
            try:
```

```
                img_array = cv2.imread(os.path.join(path,img))
```

```

        new_array = cv2.resize(img_array, (IMG_SIZE,IMG_SIZE))

        training_data.append([new_array, class_num])

    except Exception as e:

        pass

create_training_data()

```

La función `create_training_data`, genera un array con la matriz correspondiente a la imagen en RGB, y además anexa el índice de la categoría a la que corresponde dicha imagen, recorriendo todas las carpetas contenidas en la ruta local que se definió en la variable "DATADIR".

```

import random

random.shuffle(training_data)

```

Esta parte del código genera un orden aleatorio en los datos de entrenamiento.

```

X=[]

y=[]

for features, label in training_data:

    X.append(features)

    y.append(label)

```

```
X = np.array(X).reshape(-1, IMG_SIZE, IMG_SIZE, 3) # 1->grayscale
```

```
Xtraining_validation = X[27:274]
```

```
Xtesting = X[0:27]
```

```
ytraining_validation = y[27:274]
```

```
ytesting = y[0:27]
```

A partir de dichos datos de entrenamiento se separan los array en X que contiene las imágenes y Y que contiene las etiquetas de las categorías.

```
import pickle
```

```
pickle_out = open("XAvocTrainVal.pickle","wb")
```

```
pickle.dump(Xtraining_validation, pickle_out)
```

```
pickle_out.close()
```

```
pickle_out = open("XAvocTest.pickle","wb")
```

```
pickle.dump(Xtesting, pickle_out)
```

```
pickle_out.close()
```

```
labels = [0,1,2]
```

```
mlbTrainVal = label_binarize(ytraining_validation,labels)
```

```
mlbTest = label_binarize(ytesting,labels)
```

```
pickle_out = open("yTrainVal.pickle","wb")
```

```
pickle.dump(mlbTrainVal, pickle_out)
```

```
pickle_out.close()
```

```
pickle_out = open("yTest.pickle","wb")
```

```
pickle.dump(mlbTest, pickle_out)
```

```
pickle_out.close()
```

En esta parte se realiza el guardado en archivos de los datos de entrenamiento y pruebas, utilizando pickle.

Estos archivos son utilizados posteriormente en el entrenamiento y en la evaluación del clasificador.

```
import tensorflow as tf
```

```
from tensorflow.keras.datasets import cifar10
```

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```
from tensorflow.keras.models import Sequential, Model
```

```
from tensorflow.keras.layers import Dense, Dropout, Activation, Flatten
```

```
from tensorflow.keras.layers import Conv2D, MaxPooling2D

from tensorflow.keras.layers import BatchNormalization, Lambda, Input

from tensorflow.keras.layers import Convolution2D, MaxPooling2D, AveragePooling2D,
ZeroPadding2D, Reshape

from tensorflow.keras.callbacks import TensorBoard

from tensorflow.keras.optimizers import SGD

from tensorflow.keras.layers import Concatenate

from tensorflow.keras.layers import BatchNormalization

import numpy as np

import os

import pickle

import time

pickle_in = open("XAvocTrainVal.pickle","rb")

X = pickle.load(pickle_in)

pickle_in = open("yTrainVal.pickle","rb")

y = pickle.load(pickle_in)

X = X/255.0
```

```
pickle_in = open("XAvocTest.pickle","rb")
```

```
Xtest = pickle.load(pickle_in)
```

```
pickle_in = open("yTest.pickle","rb")
```

```
ytest = pickle.load(pickle_in)
```

```
NAME = "resultadosAvocadoCNNFinal"
```

```
tboard_log_dir = os.path.join("logs8",NAME)
```

```
tensorboard = TensorBoard(log_dir = tboard_log_dir)
```

```
#####
```

First

path

```
#####
```

```
#branch 1
```

```
input_image = Input(shape=X.shape[1:])
```

```
branch1
```

=

```
Conv2D(filters=48,
```

```
kernel_size=(11,11),strides=(4,4),input_shape=X.shape[1:])(input_image)
```

```
branch1 = Activation("relu")(branch1)
```

```
branch1 = BatchNormalization()(branch1)
```

```
branch1 = MaxPooling2D(pool_size=(3,3),strides=(2,2))(branch1)
```

```
#Splitting output from branch1 in two halves
```

```
branch11 = Lambda(lambda x : x[:, :, :, :24])(branch1) # branch 1.1
```

```
branch12 = Lambda(lambda x : x[:, :, :, 24:])(branch1) # branch 1.2
```

```
#branch 1.1.
```

```
branch11 = Conv2D(filters=64,  
kernel_size=(3,3),strides=(1,1),padding='same')(branch11)
```

```
branch11 = Activation("relu")(branch11)
```

```
branch11 = BatchNormalization()(branch11)
```

```
branch11 = MaxPooling2D(pool_size=(3,3), strides=(2,2))(branch11)
```

```
#branch 1.2.
```

```
branch12 = Conv2D(filters=64,  
kernel_size=(3,3),strides=(1,1),padding='same')(branch12)
```

```
branch12 = Activation("relu")(branch12)
```



```
branch12 = BatchNormalization()(branch12)
```

```
branch12 = MaxPooling2D(pool_size=(3,3), strides=(2,2))(branch12)
```

```
#Concatenating branches 1.1 and 1.2 into branch 1.3
```

```
branch13 = Concatenate()([branch11,branch12]) #branch 1.3
```

```
branch13 = Conv2D(filters=192,kernel_size=(3,3),strides=(1,1),padding='same')(branch13)
```

```
branch13 = Activation("relu")(branch13)
```

```
#Splitting branch 1.3 into two halves
```

```
branch14 = Lambda(lambda x : x[:,::2,:96])(branch13) #branch 1.4
```

```
branch15 = Lambda(lambda x : x[:,::2,96:])(branch13) #branch 1.5
```

```
#branch 1.4
```

```
branch14 = Conv2D(filters=96, kernel_size=(3,3), strides=(1,1),  
padding='same')(branch14)
```

```
branch14 = Activation("relu")(branch14)
```

#branch 1.5

```
branch15 = Conv2D(filters=96, kernel_size=(3,3), strides=(1,1),  
padding='same')(branch15)
```

```
branch15 = Activation("relu")(branch15)
```

#branch 1.6

```
branch16 = Conv2D(filters=64, kernel_size=(3,3), strides=(1,1),  
padding='same')(branch14)
```

```
branch16 = Activation("relu")(branch16)
```

```
branch16 = MaxPooling2D(pool_size=(3,3),strides=(2,2))(branch16)
```

#branch 1.7

```
branch17 = Conv2D(filters=64, kernel_size=(3,3),strides=(1,1),  
padding='same')(branch15)
```

```
branch17 = Activation("relu")(branch17)
```

```
branch17 = MaxPooling2D(pool_size=(3,3), strides=(2,2))(branch17)
```

#####

#####

#####

Second

path

#####

branch2 = Conv2D(filters=48,
kernel_size=(11,11),strides=(4,4),input_shape=X.shape[1:])(input_image)

branch2 = Activation("relu")(branch2)

branch2 = BatchNormalization()(branch2)

branch2 = MaxPooling2D(pool_size=(3,3), strides=(2,2))(branch2)

branch21 = Lambda(lambda x:x[:, :, :, :24])(branch2)

branch22 = Lambda(lambda x:x[:, :, :, 24:])(branch2)

branch21 = Conv2D(filters=64, kernel_size=(3,3),strides=(1,1),
padding='same')(branch21)

branch21 = Activation("relu")(branch21)

branch21 = BatchNormalization()(branch21)

```
branch21 = MaxPooling2D(pool_size=(3,3), strides=(2,2))(branch21)
```

```
branch22 = Conv2D(filters=64, kernel_size=(3,3),strides=(1,1),  
padding='same')(branch22)
```

```
branch22 = Activation("relu")(branch22)
```

```
branch22 = BatchNormalization()(branch22)
```

```
branch22 = MaxPooling2D(pool_size=(3,3), strides=(2,2))(branch22)
```

```
branch23 = Concatenate()([branch21,branch22])
```

```
branch23 = Conv2D(filters=192,kernel_size=(3,3),strides=(1,1),padding='same')(branch23)
```

```
branch23 = Activation("relu")(branch23)
```

```
branch24 = Lambda(lambda x : x[:, :, :, :96])(branch23)
```

```
branch25 = Lambda(lambda x : x[:, :, :, 96:])(branch23)
```

```
branch24 = Conv2D(filters=96, kernel_size=(3,3), strides=(1,1),  
padding='same')(branch24)
```

```
branch24 = Activation("relu")(branch24)
```

```
branch25 = Conv2D(filters=96, kernel_size=(3,3), strides=(1,1),  
padding='same')(branch25)
```

```
branch25 = Activation("relu")(branch25)
```

```
branch26 = Conv2D(filters=64, kernel_size=(3,3), strides=(1,1),  
padding='same')(branch24)
```

```
branch26 = Activation("relu")(branch26)
```

```
branch26 = MaxPooling2D(pool_size=(3,3),strides=(2,2))(branch26)
```

```
branch27 = Conv2D(filters=64, kernel_size=(3,3),strides=(1,1),  
padding='same')(branch25)
```

```
branch27 = Activation("relu")(branch27)
```

```
branch27 = MaxPooling2D(pool_size=(3,3), strides=(2,2))(branch27)
```

```
merged_branches = Concatenate()([branch16,branch17,branch26,branch27])
```

```
flatten = Flatten()(merged_branches)
```

```
FC_1 = Dense(units=4096, activation='relu')(flatten)
```

```
FC_1 = Dropout(0.6)(FC_1)
```

```
FC_2 = Dense(units=4096, activation='relu')(FC_1)
```

```
FC_2 = Dropout(0.6)(FC_2)
```

```
output = Dense(units=3, activation='softmax')(FC_2)
```

```
model = Model(inputs=input_image, outputs=output)
```

```
sgd = SGD(lr=1e-3, decay=1e-6, momentum=0.9, nesterov=True)
```

```
model.compile(optimizer=sgd, loss='categorical_crossentropy', metrics=['accuracy'])
```

```
model.fit(X,y,          batch_size=32,          epochs=74,          validation_split=0.3,  
callbacks=[tensorboard])
```

```
results = model.predict(Xtest)
```

En el bloque de código anterior se observa la arquitectura de la red neuronal implementada, la cual al final del mismo, genera unos resultados predictivos que son almacenados en la variable “results”.

Luego de esto, existen dos caminos: si se desea evaluar las predicciones del modelo, o si simplemente se desea conocer la predicción del este.

Para el primer caso, se tiene el siguiente bloque de código, el cual mostrará la imagen del aguacate, junto con la categoría predicha, y la categoría original:

```
from PIL import Image

from matplotlib import pyplot as plt

correctos = 0

CATEGORIES = ["exportation", "matures", "nationals"]

for i in range(0, len(results)):

    plt.imshow(Xtest[i], interpolation='nearest')

    plt.show()

    print("valor clasificado: ")

    print(CATEGORIES[np.argmax(results[i])])

    print("valor correcto")

    print(CATEGORIES[np.argmax(ytest[i])])
```

En el segundo caso, es decir, cuándo sólo se quiere saber la predicción realizada, se tiene el siguiente bloque de código:

```
from PIL import Image

from matplotlib import pyplot as plt

correctos = 0

CATEGORIES = ["exportation", "matures", "nationals"]

for i in range(0, len(results)):

    plt.imshow(Xtest[i], interpolation='nearest')

    plt.show()

    print("valor clasificado: ")

    print(CATEGORIES[np.argmax(results[i])])
```


RESULTADOS

De acuerdo con la experimentación realizada, en donde a partir de las imágenes adquiridas en tres categorías distintas de maduración comercial, se realiza el entrenamiento de la red neuronal convolucional en 300 ciclos completos de entrenamiento, escogiendo entre sus múltiples resultados, aquellos que presentan los resultados que se ajustan mejor a un nivel de precisión y generalización aceptables para el problema.

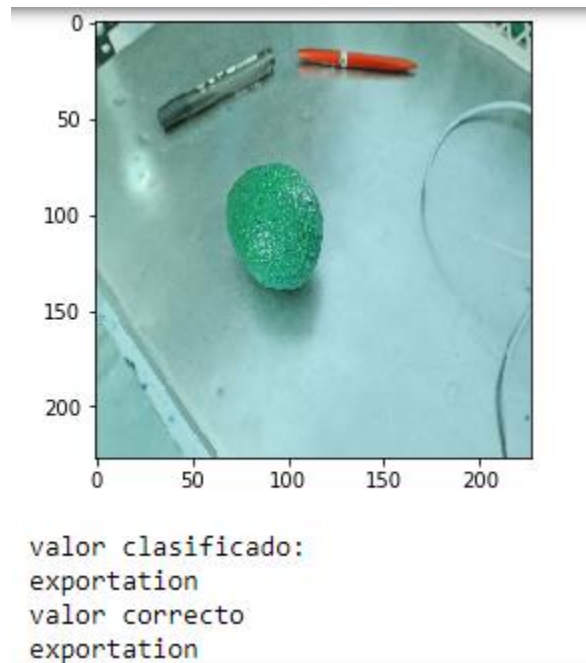


Ilustración 7 Ejemplo de clasificación correcta de aguacate tipo exportación

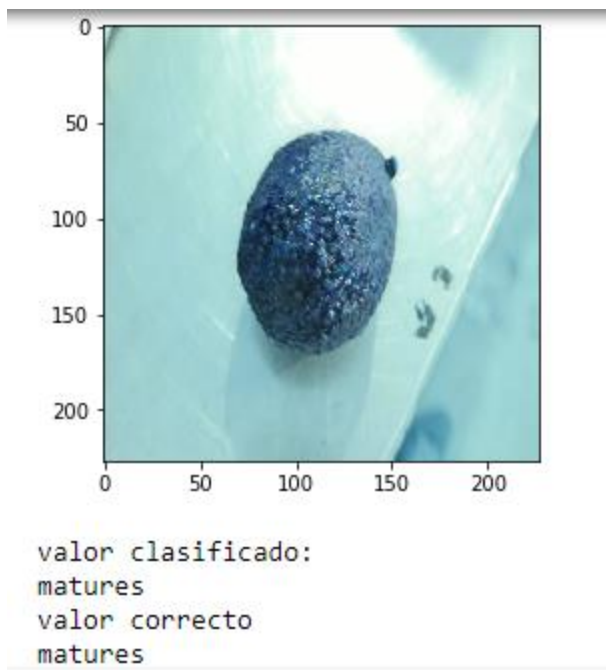


Ilustración 8 Ejemplo de clasificación correcta de aguacate tipo maduros

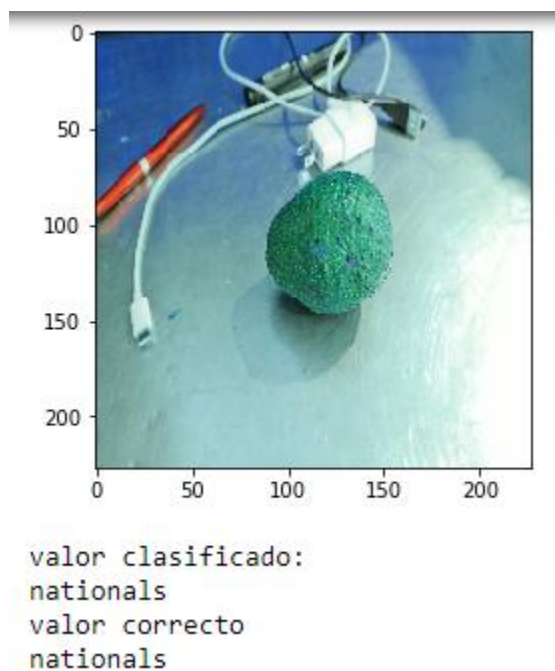


Ilustración 9 Ejemplo de clasificación correcta de aguacate tipo nacional

Los siguientes resultados obtenidos corresponden a pruebas realizadas sobre un espacio muestral de 248 aguacates en 3 categorías de madurez comercial de aguacates Hass, se encuentra un nivel de precisión de 96.05 % luego de 72 ciclos de entrenamientos, en los resultados de validación.

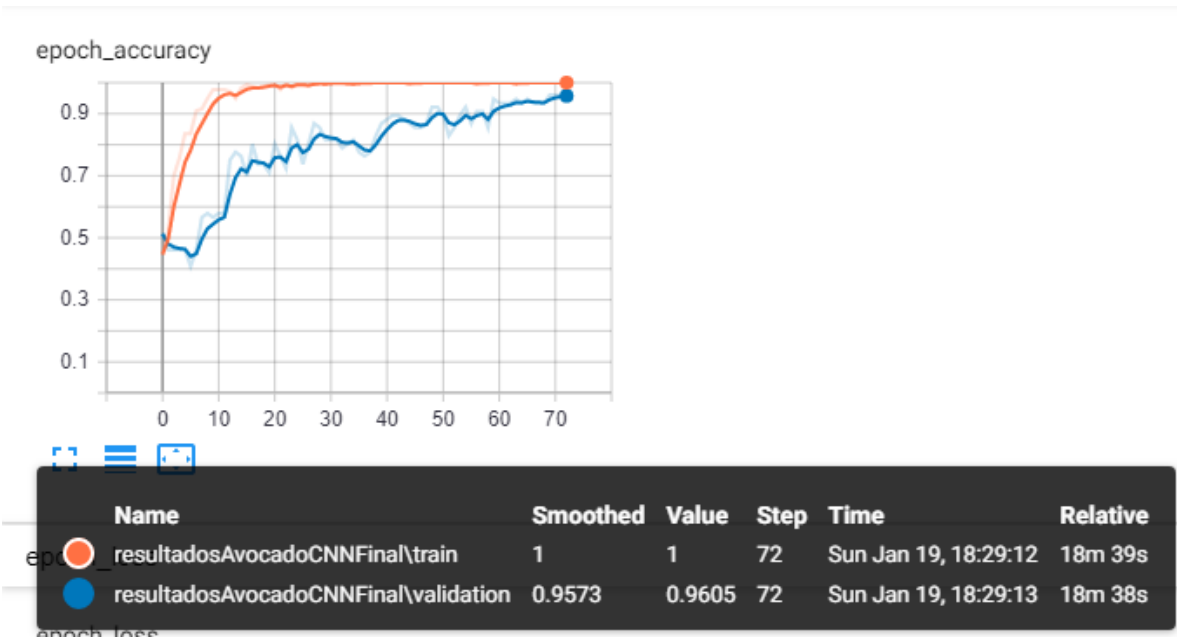


Ilustración 10: precisión de la arquitectura en 72 ciclos de entrenamiento.

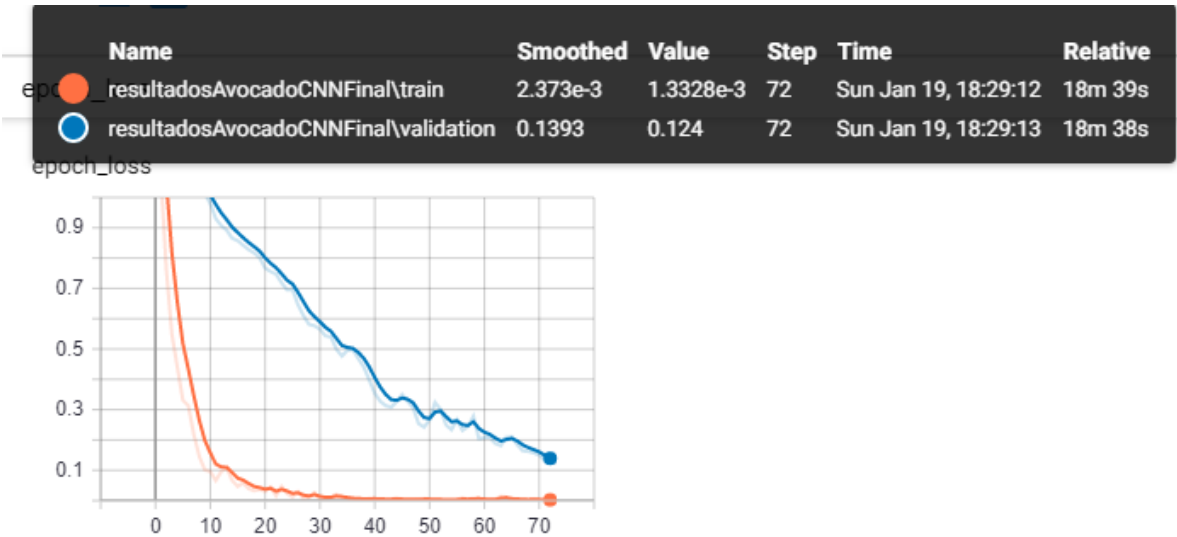


Ilustración 11: valores de pérdida en el ciclo 72 de entrenamiento

Se aclara que se toma este ciclo de entrenamiento en particular, en un espacio de 300 ciclos de entrenamiento, dado que se considera que es un buen nivel de aprendizaje, dado que corresponde a un valor del 95.73% de precisión, y que puede ser lo suficientemente general al momento de realizar la clasificación, ya que no presenta overfitting.

Así mismo, se utilizan 27 imágenes aparte del espacio muestral que fue designado para entrenamiento y validación, como herramienta para evaluar al modelo obteniendo los siguientes resultados en la siguiente matriz de confusión:

		Valores predichos		
		Exportación	Maduros	Nacionales
Valores reales	Exportación	14	0	0
	Maduros	0	4	1
	Nacionales	1	0	7

Tabla 1: matriz de confusión a partir de los resultados obtenidos

La tabla anterior representa las clasificaciones obtenidas a partir de las categorías reales de las imágenes que fueron utilizadas para evaluar el clasificador.

Así mismo, se obtienen los siguientes valores a partir de dicha información:

Clase	Recall	Precision	F-measure	Macro F-measure
Exportación	100,00%	93,33%	96,55%	90,98%
Maduros	80,00%	100,00%	88,89%	
Nacionales	87,50%	87,50%	87,50%	

Tabla 2: valores obtenidos a partir de la matriz de confusión

En la tabla 2, se obtiene un valor de 80% o más para la columna de Recall o sensibilidad, la cual expresa el porcentaje de imágenes que perteneciendo a una clasificación fueron correctamente predichas como miembros de esa clasificación, y para la precisión se obtuvo un valor con un mínimo de 87,50%, el cual indica la proporción entre las clasificaciones correctas que ha realizado el clasificador y el total de predicciones.

Y por último, se obtiene el F-measure, el cual es el valor estadístico correspondiente al balanceo entre el Recall y la Precisión, el cuál nuevamente representa valores superiores a 87%, y al ser promediados, se obtiene un valor de 90,98%.

CONCLUSIONES

- Se realiza la experimentación sobre ambientes de alto ruido con un espacio muestral de 275 aguacates Hass, (que fue dividido en 248 imágenes para entrenamiento y pruebas, 27 para validación de la clasificación), obteniendo un valor estadístico F1 de 90.98%.
- Se realiza una propuesta con respecto a la detección simultánea del grado de maduración de los aguacates, la cual corresponde a un nuevo proceso operativo que permite la generación de conjuntos de fotografías para ser analizadas.
- Los valores promedio del valor estadístico F1, para la totalidad de las clases es alto, lo cual reafirma a esta solución arquitectónica como válida ante el problema que se pretendía solucionar.
- Se concluye que es posible realizar la clasificación del grado de madurez del aguacate, utilizando redes neuronales convolucionales.
- Se concluye que es por medio del clasificador propuesto, se pueden clasificar imágenes de aguacates Hass, en ambientes no controlados.

ANEXOS

Es necesario resaltar que el código fuente que se presentará a continuación, hace parte del desarrollo investigativo de R. F. Rachmadi y I K. E. Purnama, a partir del cual se realizó este prototipo.

1. Script para realizar la generación del conjunto de entrenamiento, validación y testing:

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
import os
```

```
import cv2
```

```
from sklearn.preprocessing import label_binarize
```

```
DATADIR = "C:/Users/alejo/proyecto/avocados/"
```

```
CATEGORIES = ["exportation", "matures", "nationals"]
```

```
IMG_SIZE = 227
```

```
training_data = []
```

```

def create_training_data():

    for category in CATEGORIES:

        path = os.path.join(DATADIR,category)

        class_num = CATEGORIES.index(category)

        for img in os.listdir(path):

            try:

                img_array = cv2.imread(os.path.join(path,img))

                new_array = cv2.resize(img_array, (IMG_SIZE,IMG_SIZE))

                training_data.append([new_array, class_num])

            except Exception as e:

                pass

create_training_data()

import random

random.shuffle(training_data)

X=[]

y=[]

```



```
for features, label in training_data:
```

```
    X.append(features)
```

```
    y.append(label)
```

```
X = np.array(X).reshape(-1, IMG_SIZE, IMG_SIZE, 3) # 1->grayscale
```

```
Xtraining_validation = X[27:274]
```

```
Xtesting = X[0:27]
```

```
ytraining_validation = y[27:274]
```

```
ytesting = y[0:27]
```

```
import pickle
```

```
pickle_out = open("XAvocTrainVal.pickle", "wb")
```

```
pickle.dump(Xtraining_validation, pickle_out)
```

```
pickle_out.close()
```

```
pickle_out = open("XAvocTest.pickle","wb")
```

```
pickle.dump(Xtesting, pickle_out)
```

```
pickle_out.close()
```

```
labels = [0,1,2]
```

```
mlbTrainVal = label_binarize(ytraining_validation,labels)
```

```
mlbTest = label_binarize(ytesting,labels)
```

```
pickle_out = open("yTrainVal.pickle","wb")
```

```
pickle.dump(mlbTrainVal, pickle_out)
```

```
pickle_out.close()
```

```
pickle_out = open("yTest.pickle","wb")
```

```
pickle.dump(mlbTest, pickle_out)
```

```
pickle_out.close()
```

2. Modelo de clasificación:

```
import tensorflow as tf
```

```
from tensorflow.keras.datasets import cifar10

from tensorflow.keras.preprocessing.image import ImageDataGenerator

from tensorflow.keras.models import Sequential, Model

from tensorflow.keras.layers import Dense, Dropout, Activation, Flatten

from tensorflow.keras.layers import Conv2D, MaxPooling2D

from tensorflow.keras.layers import BatchNormalization, Lambda, Input

from tensorflow.keras.layers import Convolution2D, MaxPooling2D, AveragePooling2D,
ZeroPadding2D, Reshape

from tensorflow.keras.callbacks import TensorBoard

from tensorflow.keras.optimizers import SGD

from tensorflow.keras.layers import Concatenate

from tensorflow.keras.layers import BatchNormalization

import numpy as np

import os

import pickle

import time


pickle_in = open("XAvocTrainVal.pickle","rb")

X = pickle.load(pickle_in)
```

```
pickle_in = open("yTrainVal.pickle","rb")
```

```
y = pickle.load(pickle_in)
```

```
X = X/255.0
```

```
pickle_in = open("XAvocTest.pickle","rb")
```

```
Xtest = pickle.load(pickle_in)
```

```
pickle_in = open("yTest.pickle","rb")
```

```
ytest = pickle.load(pickle_in)
```

```
NAME = "resultadosAvocadoCNNFinal"
```

```
tboard_log_dir = os.path.join("logs6",NAME)
```

```
tensorboard = TensorBoard(log_dir = tboard_log_dir)
```

```
##### First path
```

```
#branch 1
```

```
input_image = Input(shape=X.shape[1:])
```

```
branch1 = Conv2D(filters=48,  
kernel_size=(11,11),strides=(4,4),input_shape=X.shape[1:])(input_image)
```

```
branch1 = Activation("relu")(branch1)
```

```
branch1 = BatchNormalization()(branch1)
```

```
branch1 = MaxPooling2D(pool_size=(3,3),strides=(2,2))(branch1)
```

```
#Splitting output from branch1 in two halves
```

```
branch11 = Lambda(lambda x : x[:,::2,:24])(branch1) # branch 1.1
```

```
branch12 = Lambda(lambda x : x[:,::2,24:])(branch1) # branch 1.2
```

```
#branch 1.1.
```

```
branch11 = Conv2D(filters=64,  
kernel_size=(3,3),strides=(1,1),padding='same')(branch11)
```

```
branch11 = Activation("relu")(branch11)
```

```
branch11 = BatchNormalization()(branch11)
```

```
branch11 = MaxPooling2D(pool_size=(3,3), strides=(2,2))(branch11)
```

```
#branch 1.2.
```

```
branch12 = Conv2D(filters=64,  
kernel_size=(3,3),strides=(1,1),padding='same')(branch12)
```

```
branch12 = Activation("relu")(branch12)
```

```
branch12 = BatchNormalization()(branch12)
```

```
branch12 = MaxPooling2D(pool_size=(3,3), strides=(2,2))(branch12)
```

```
#Concatenating branches 1.1 and 1.2 into branch 1.3
```

```
branch13 = Concatenate()([branch11,branch12]) #branch 1.3
```

```
branch13 =  
Conv2D(filters=192,kernel_size=(3,3),strides=(1,1),padding='same')(branch13)
```

```
branch13 = Activation("relu")(branch13)
```

```
#Splitting branch 1.3 into two halves
```

```
branch14 = Lambda(lambda x : x[:, :, :, :96])(branch13) #branch 1.4
```

```
branch15 = Lambda(lambda x : x[:, :, :, 96:])(branch13) #branch 1.5
```

#branch 1.4

```
branch14 = Conv2D(filters=96, kernel_size=(3,3), strides=(1,1),  
padding='same')(branch14)
```

```
branch14 = Activation("relu")(branch14)
```

#branch 1.5

```
branch15 = Conv2D(filters=96, kernel_size=(3,3), strides=(1,1),  
padding='same')(branch15)
```

```
branch15 = Activation("relu")(branch15)
```

#branch 1.6

```
branch16 = Conv2D(filters=64, kernel_size=(3,3), strides=(1,1),  
padding='same')(branch14)
```

```
branch16 = Activation("relu")(branch16)
```

```
branch16 = MaxPooling2D(pool_size=(3,3),strides=(2,2))(branch16)
```

#branch 1.7

```

branch17 = Conv2D(filters=64, kernel_size=(3,3),strides=(1,1),
padding='same')(branch15)

branch17 = Activation("relu")(branch17)

branch17 = MaxPooling2D(pool_size=(3,3), strides=(2,2))(branch17)

```

Second path

```

branch2 = Conv2D(filters=48,
kernel_size=(11,11),strides=(4,4),input_shape=X.shape[1:])(input_image)

```

```

branch2 = Activation("relu")(branch2)

```

```

branch2 = BatchNormalization()(branch2)

```

```

branch2 = MaxPooling2D(pool_size=(3,3), strides=(2,2))(branch2)

```

```

branch21 = Lambda(lambda x:x[:, :, :, :24])(branch2)

```

```

branch22 = Lambda(lambda x:x[:, :, :, 24:])(branch2)

```

```

branch21 = Conv2D(filters=64, kernel_size=(3,3),strides=(1,1),
padding='same')(branch21)

```

```

branch21 = Activation("relu")(branch21)

```

```

branch21 = BatchNormalization()(branch21)

```



```
branch21 = MaxPooling2D(pool_size=(3,3), strides=(2,2))(branch21)
```

```
branch22 = Conv2D(filters=64, kernel_size=(3,3),strides=(1,1),  
padding='same')(branch22)
```

```
branch22 = Activation("relu")(branch22)
```

```
branch22 = BatchNormalization()(branch22)
```

```
branch22 = MaxPooling2D(pool_size=(3,3), strides=(2,2))(branch22)
```

```
branch23 = Concatenate()([branch21,branch22])
```

```
branch23 =  
Conv2D(filters=192,kernel_size=(3,3),strides=(1,1),padding='same')(branch23)
```

```
branch23 = Activation("relu")(branch23)
```

```
branch24 = Lambda(lambda x : x[:, :, :, :96])(branch23)
```

```
branch25 = Lambda(lambda x : x[:, :, :, 96:])(branch23)
```

```
branch24 = Conv2D(filters=96, kernel_size=(3,3), strides=(1,1),  
padding='same')(branch24)
```

```
branch24 = Activation("relu")(branch24)
```

```
branch25 = Conv2D(filters=96, kernel_size=(3,3), strides=(1,1),  
padding='same')(branch25)
```

```
branch25 = Activation("relu")(branch25)
```

```
branch26 = Conv2D(filters=64, kernel_size=(3,3), strides=(1,1),  
padding='same')(branch24)
```

```
branch26 = Activation("relu")(branch26)
```

```
branch26 = MaxPooling2D(pool_size=(3,3),strides=(2,2))(branch26)
```

```
branch27 = Conv2D(filters=64, kernel_size=(3,3),strides=(1,1),  
padding='same')(branch25)
```

```
branch27 = Activation("relu")(branch27)
```

```
branch27 = MaxPooling2D(pool_size=(3,3), strides=(2,2))(branch27)
```

```
merged_branches = Concatenate()([branch16,branch17,branch26,branch27])
```

```
flatten = Flatten()(merged_branches)
```

```
FC_1 = Dense(units=4096, activation='relu')(flatten)
```

```
FC_1 = Dropout(0.6)(FC_1)
```

```
FC_2 = Dense(units=4096, activation='relu')(FC_1)
```

```
FC_2 = Dropout(0.6)(FC_2)
```

```
output = Dense(units=3, activation='softmax')(FC_2)
```

```
model = Model(inputs=input_image, outputs=output)
```

```
sgd = SGD(lr=1e-3, decay=1e-6, momentum=0.9, nesterov=True)
```

```
model.compile(optimizer=sgd, loss='categorical_crossentropy', metrics=['accuracy'])
```

```
model.fit(X,y, batch_size=32, epochs=74, validation_split=0.3,  
callbacks=[tensorboard])
```

3. Luego de tener el modelo entrenado, se debe de ejecutar lo siguiente:

```
results = model.predict(Xtest)
```

TRABAJOS FUTUROS

- Como una posible alternativa arquitectónica se propone la implementación de una arquitectura que use máquinas de soporte vectorial para la clasificación del grado de madurez de los aguacates tipo Hass.
- Se propone a su vez la implementación de un sistema completo que permita la implantación en una empresa de exportación de aguacates Hass, del clasificador actualmente desarrollado.
- Se propone la creación de una etapa de preprocesamiento de imágenes en donde los ruidos como la luminosidad puedan ser reducidos, para que estos no interfieran en los posibles resultados emitidos por el clasificador.
- Se propone además un estudio estadístico sobre cómo los distintos ruidos en la adquisición de las imágenes pueden afectar los resultados obtenidos.

BIBLIOGRAFÍA

- [1] Ministerio colombiano de agricultura, 2019, “Cadena de aguacate, indicadores e instrumentos”.
- [2] Y. LeCun, Y. Bengio, G. Hinton, “Deep learning”, Nature Research Journal, 2015
- [3] K. Gurney, “An introduction to neural network”, 1997, University College London.
- [4] W. E. Choque Valderrama, C. C. Salazar Salazar, 2016 ,“Desarrollo de algoritmo y prototipo móvil para medir el grado de madurez del aguacate Hass mediante procesamiento digital de imágenes”, Universidad Tecnológica de Pereira.
- [5] F. M. A. Mazen, A. A. Nashat, 2019, “Ripeness Classification of Bananas Using an Artificial Neural Network”, Springer.
- [6] I. Arzate-Vázquez, J. Chanona-Pérez, M.J. Perea-Flores, G. Calderón-Dominguez ,“Image Processing Applied to Classification of Avocado Variety Hass (Persea amaricana Mill.) During the Ripening Process”, Springer, volumen 4, 2011.
- [7] Jun-Wei Hsieh, Li-Chih Chen, Sin-Yu Chen, Duan-Yu Chen, Salah Alghyaline, Hui-Fen Chiang,“Vehicle Color Classification Under Different Lighting Conditions Through Color Correction”, IEEE Sensors journal, 2015
- [8] R. F. Rachmadi, I K. E. Purnama ,“Vehicle Color Recognition using Convolutional Neural Network”, Cornell University, 2018

- [9] R. Vigneshwar, Ms. V. Prema, "Colour Recognition in Images Using Neural Networks", International Journal of Innovative Research in Computer and Communication Engineering, 2016
- [10] M. Islam, K. Wahid, "Assessment of Ripening Degree of Avocado by Electrical Impedance Spectroscopy and Support Vector Machine", Journal of Food Quality, 2018.
- [11] K. Bhoyar, O. Kakde, "Skin Color Detection Model Using Neural Networks and its Performance Evaluation", Journal of Computer Science, 2010.
- [12] M. Mora, J. Aliaga, C. Fredes, "Olive Ripening Phase Estimation based on Neural Network", Congreso Argentino de AgroInformática, 2017.
- [13] Y. Wang, J. Liu, J. Wang, Y. Li, H. Lu, "Color names learning using convolutional neural networks", IEEE International Conference on Image Processing, 2015
- [14] Y. Zhang, J. Lian, M. Fan, Y. Zheng, "Deep indicator for fine-grained classification of banana's ripening stages", EURASIP Journal on Image and Video Processing, 2018
- [15] J. Wu, B. Zhang, J. Zhou, Y. Xiong, B. Gu, Y. Xiaolong, "Automatic Recognition of Ripening Tomatoes by Combining Multi-Feature Fusion with a Bi-Layer Classification Strategy for Harvesting Robots", Sensors, 2019
- [16] A. Sanaeifar, A. Bakhshipour, M. Guardia, "Prediction of banana quality indices from color features using support vector regression", Talanta, volumen 148, 2015.
- [17] S. K. Gurupatham, E. Ilksoy, N. Jacob, K. Horn, F. Fahad, "Fruit Ripeness Estimation for Avocado Using Thermal Imaging", The American Society of Mechanical Engineers, 2018.

- [18] R. Hamza, M. Chtourou, "Apple Ripeness Estimation Using Artificial Neural Network", IEEE International Conference on High Performance Computing (HPCS) & Simulation, 2018
- [19] W. D. Niño Pacheco, F. R. Jiménez López, "Tomato classification according to organoleptic maturity (coloration) using machine learning algorithms K-NN, MLP, and K-Means Clustering", IEEE XXII Symposium on Image, Signal Processing and Artificial Vision, 2019
- [20] "An introduction to convolutional neural networks", 2015, Keiron Teilo O'Shea Ryan Nash, Springer.
- [21] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, Xindong Wu , "Object detection with Deep learning: A review", IEEE Transactions on neural networks and learning systems, 2019.
- [22] Naiyang Deng, Yingjie Tian, Chunhua Zhang , "Support Vector Machines Optimization Based Theory, Algorithms, and Extensions", Taylor & Francis Group, 2013.
- [23] "NORMA DEL CODEX PARA EL AGUACATE" [Online] Available: <http://www.colpos.mx/bancodenormas/ninternacionales/CODEX-STAN-197-1995.pdf>